Network Encryption

Information Assurance

Lesson Objectives

- Identify network attack methods.
- Describe how encryption works.
- Identify how keys are distributed securely over networks.
- Describe Certificate Authorities.
- Explain conventional encryption techniques used by COE Army systems.
- Explain the role of the CDAP.

Outline

- Attacks
- Network Traffic
  - Link Encryption
  - End-to-end Encryption
  - Key Management
- Certificate Authorities
- Encryption Algorithms
- Encryption v. Password Strength
- Computer Network Defense

Network Encryption

- Attacks
- Network Traffic
  - Link Encryption
  - End-to-end Encryption
  - Key Management
- Certificate Authorities
- Encryption Algorithms
- Encryption v. Password Strength
- Computer Network Defense

Anatomy of an Attack (1 of 2)

Objective	Methodologies	Techniques
Information gathering	Footprinting	Open source search, whois
Target assessment	Scanning	Ping sweep, port scan, OS detection
Intrusive probing	Enumeration	List user accounts, file shares, identify applications
Access the target	Gaining access	Password cracking, buffer overflows
Gain full access	Escalating Privilege	Password cracking, known exploits

Source: Hacking Exposed – Network Security Secrets and Solutions: Joel, Stuart, George

Signalcorps.org
footprinting

Anatomy of an Attack (2 of 2)

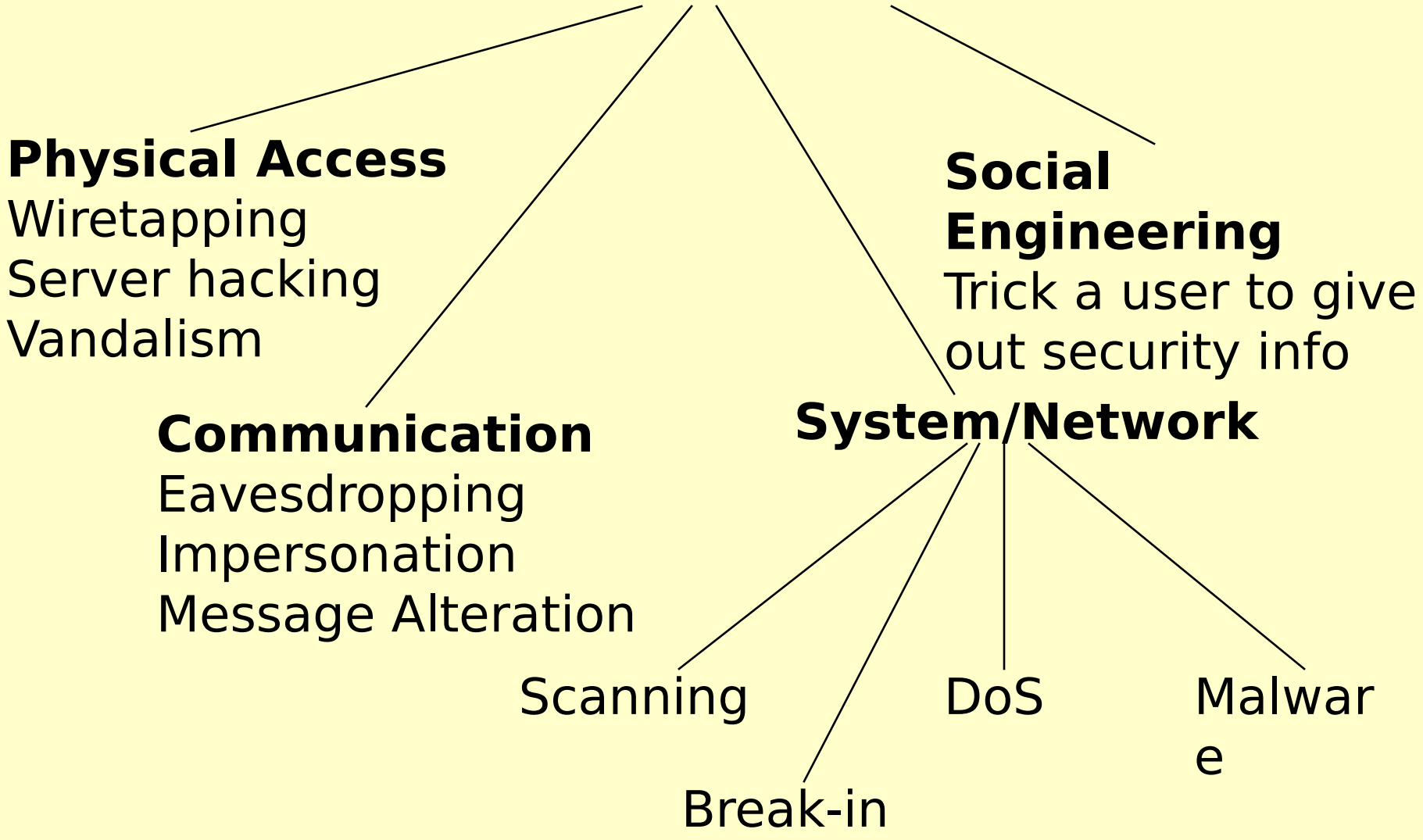
Objective	Methodologies	Techniques
Gain access to trusted host	Pilfering	Evaluate trusts, cleartext passwords
Hide footprints	Covering tracks	Clear logs
Anytime access	Creating back-doors	Plant remote control services, replace applications with trojans
If unsuccessful, run exploit	Denial-of-Service	Syn flood, ICMP Techniques, DDoS

Source: Hacking Exposed – Network Security Secrets and Solutions: Joel, Stuart, George

Attacks on Network Traffic

- Passive Methods (Packet Sniffing)
  - Reading (Content Analysis)
  - Interception or Tracking (Traffic Analysis)
  - Counter by Encryption
- Active Methods
  - “Jamming” or Interruption
  - Modification or Packet Substitution Counter by Checking Integrity
  - Spoofing or Fabrication Counter by Authentication
  - Denial of Service

Attacks on Network Traffic



Network Encryption

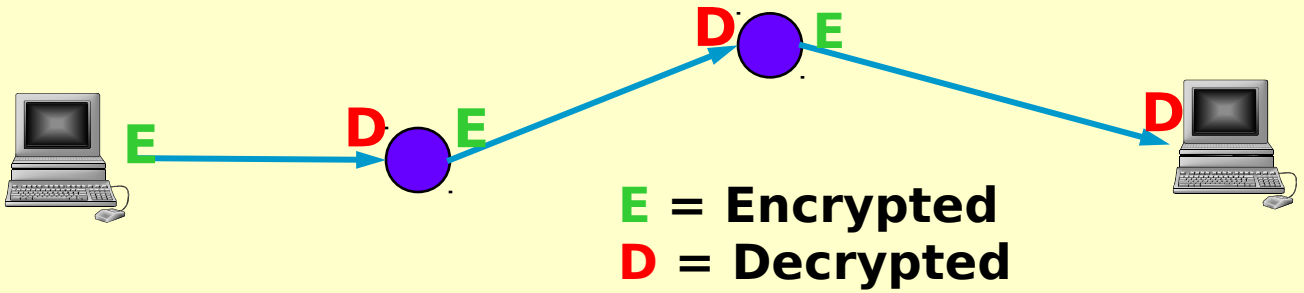
- Attacks
- Network Traffic
  - Link Encryption
  - End-to-end Encryption
  - Key Management
- Certificate Authorities
- Encryption Algorithms
- Encryption v. Password Strength
- Computer Network Defense

Basic Strategies

- Link encryption
  - Node to node
  - Based on network scheme
- End-to-end encryption
  - Host to host
  - VPN (e.g. firewall to firewall)
  - IPSEC

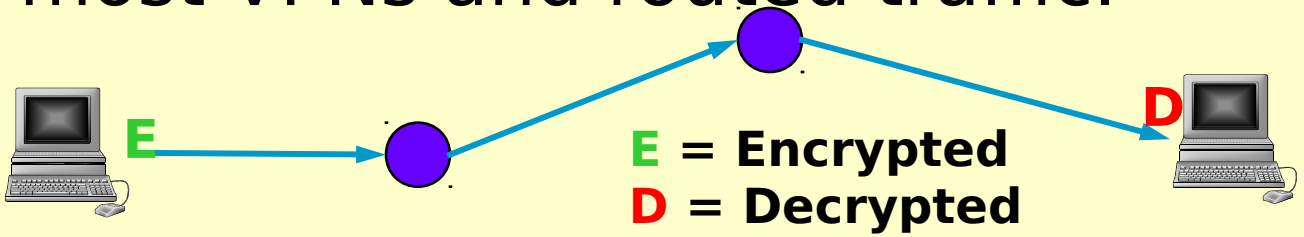
Link Encryption

- Encrypts traffic entering each link.
- Decrypts traffic leaving each link and entering a node.
- Multiple keys; slow.
- Plain-text traffic potentially vulnerable at communications nodes.
- IE: TED.



End-to-End Encryption

- Encrypts traffic at source.
  - host to host
  - firewall to firewall
- Decrypts traffic at final destination.
- Plain-text traffic not available at nodes.
- Header in plaintext; more information exposed, key more exposed.
- IE: most VPNs and routed traffic.



Key Management

- Key generation
- Key distribution
- Key change
- Key disposition
- Key recovery
- Control of cryptography keys

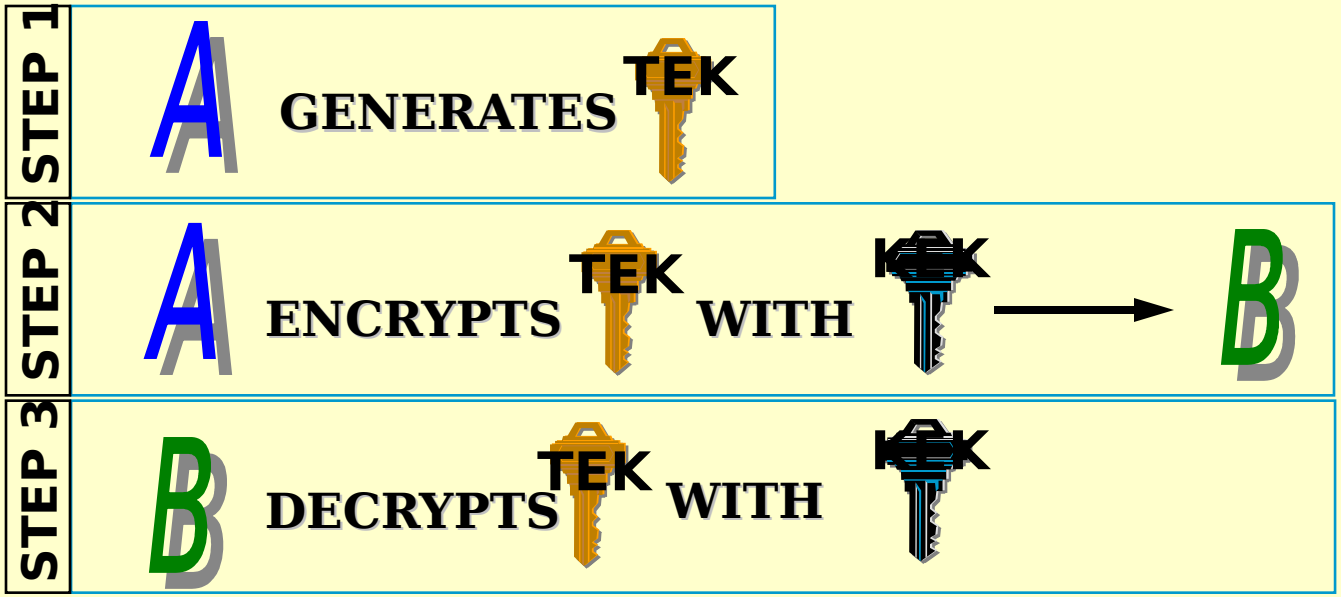
Key Management Principles

- Automated
- Integrated
- Separate KEKs
- Disguise cleartext patterns
- Infrequent use of keys with long lives
- Trusted hardware
- One copy
- Multiparty recovery controls

Key Distribution

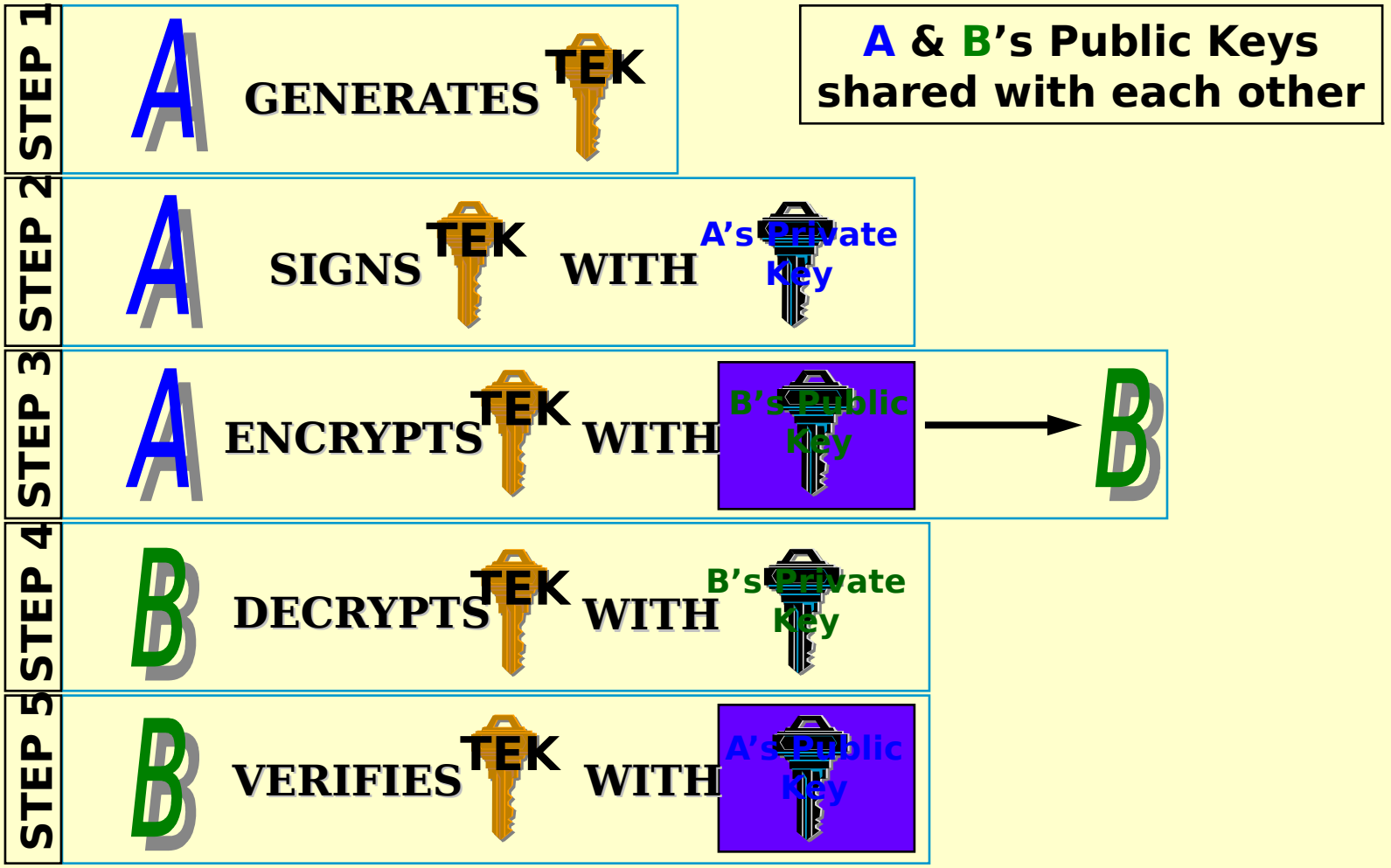
- **Manual**
  - Common in symmetric encryption.
  - KYK-13 or ANCD.
- **Automatic**
  - Encrypt with another key.
  - Authentication of source is a bonus of using public keys
  - Distribution normally from a key server or key distribution center (KDC).

Point-to-Point Using Secret Key



A & B already share a common key encryption key (KEK)

Point-to-Point Using Public Key



Network Encryption

- Attacks
- Network Traffic
  - Link Encryption
  - End-to-end Encryption
  - Key Management
- Certificate Authorities
- Encryption Algorithms
- Encryption v. Password Strength
- Computer Network Defense

Certificate Authorities (1 of 3)

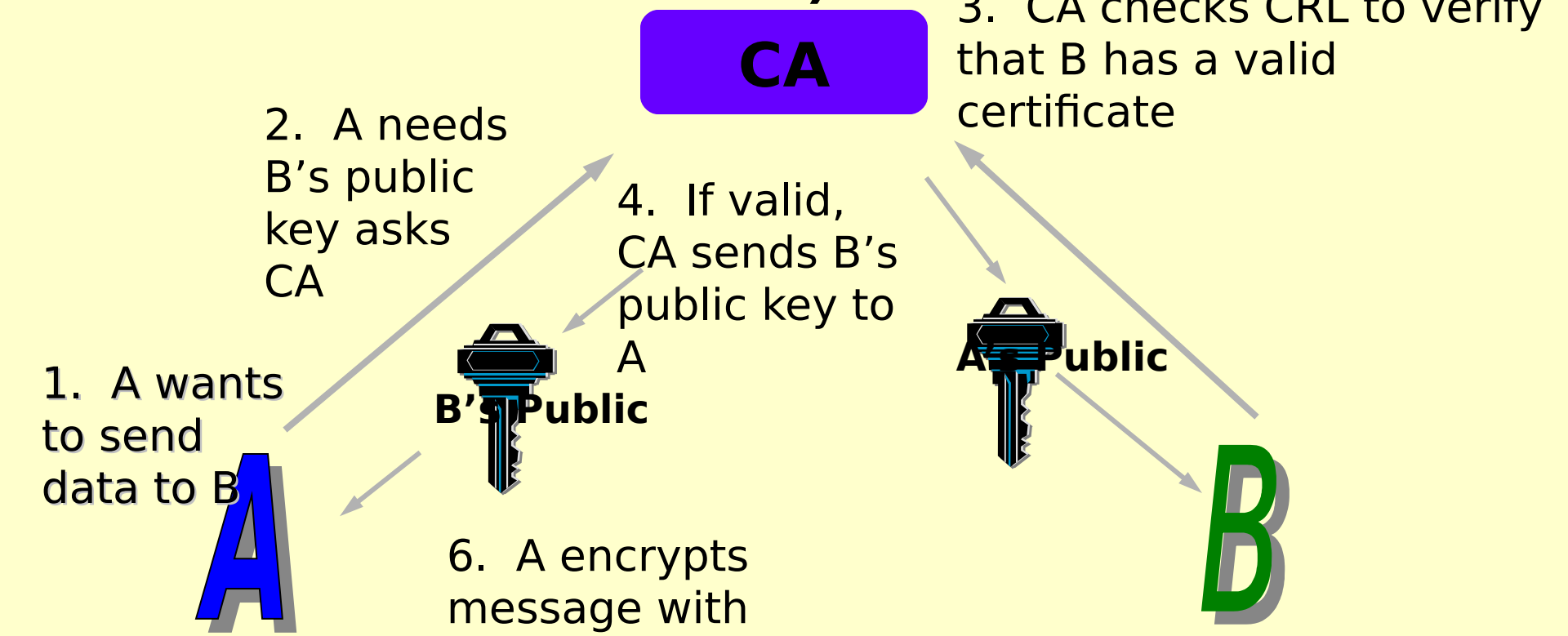
- Public keys need to be manageable
  - allow for recovery agents
  - protected by authentication and integrity services.
  - identify those that are no longer valid
- Assigns a combination of identifier and public key (certificate) for each entity in its domain.
- Private key must be protected
- Certificate Revocation List (CRL)

Certificate Authorities (2 of 3)

Windows 2000 Active Directory:

- Root CA
- Subordinate CA
- Private keys stored in an encrypted folder called …/RSA – this cannot be moved (crypto API)
- CRLs list public keys that are no longer valid.

Certificate Authorities (3 of 3)


CA
3. CA checks CRL to verify that B has a valid certificate
2. A needs B's public key asks CA
4. If valid, CA sends B's public key to A
1. A wants to send data to B
B's Public
A's Public
A
B
6. A encrypts message with B's Public Key
5. A digitally signs message with A's private key
7. Encrypted Message
8. B decrypts message with B's private key, verifies signature with A's public key.

Encryption

- Attacks
- Network Traffic
  - Link Encryption
  - End-to-end Encryption
  - Key Management
- Certificate Authorities
- Encryption Algorithms
- Encryption v. Password Strength
- Computer Network Defense

Symmetric Encryption Algorithms

- Trunk Encryption Device (TED)
- Data Encryption Standard (DES)
- Double/Triple DES
- Advanced Encryption Standard (AES) (Rijndael)
- International Data Encryption Algorithm (IDEA)
- RC6
- Blowfish

Trunk Encryption Devices (TEDs)

- Use 128 bit encryption.
- Utilize limited transposition.
- Very secure, but limited authentication.
- TEDs operate at the trunk level.
- High data rates, full duplex

Data Encryption Standard (DES)

- Uses a 56 bit key on 64 bit blocks.
- Used by several O/S to encrypt password files.
- An old business standard.
- In Jan 1999 a $250,000 computer broke DES in 22 hours, 15 minutes
  - 2004 equivalent number of current processors -probably minutes
- 5 Modes
  - Block: ECB, CBC
  - Stream: CFB, OFB, CTR

Double Data Encryption Standard (2DES)

- 112 bit key (56 bits X 2)
- 2 stages of DES with separate keys for each stage
- Based on DES - trusted cipher.
- Turned out to be no more secure than DES

Triple Data Encryption Standard (3DES)

- 168 bit key (56 bits X 3)
- 3 stages of DES with separate keys for each stage
- Based on DES - trusted cipher.
- Much faster than public key cryptography methods
- Transfer of secret key unsafe.
- 3 modes: EEE3, EDE3, EEE2 or EDE2

Advanced Encryption Standard (AES) (Rijndael)

- FIPS 197 - standard since 2001 replaces DES
- Supports block and three key sizes: 128, 192 and 256 bits.
- If DES cracking machine took 1 second, it would take 149,000,000,000,000 years to crack the 128 bit AES.
- Can run on hardware, software, browsers, smart cards (very efficient)
- Variable number of rounds, each round has 4 steps

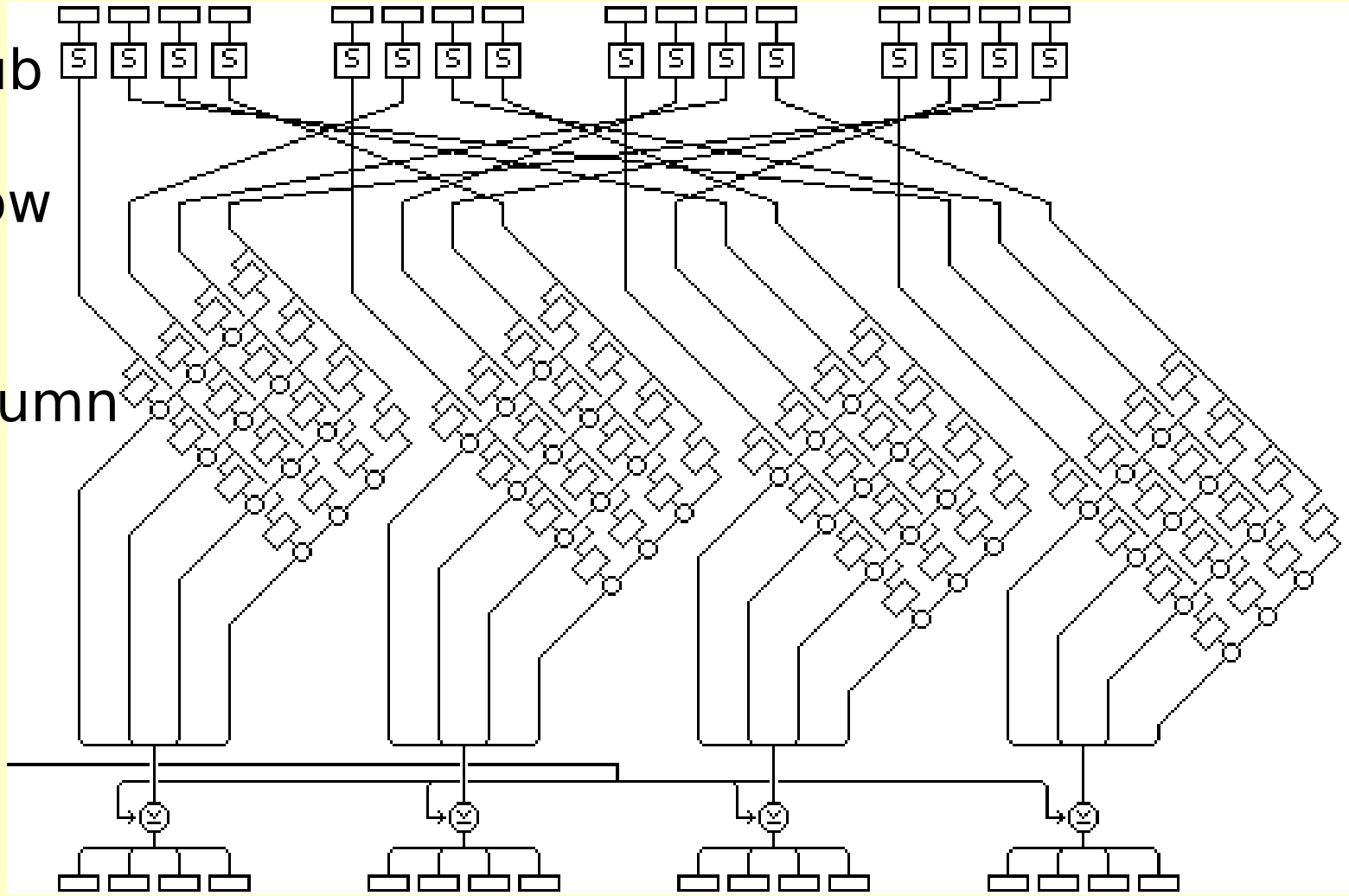
AES

Byte Sub

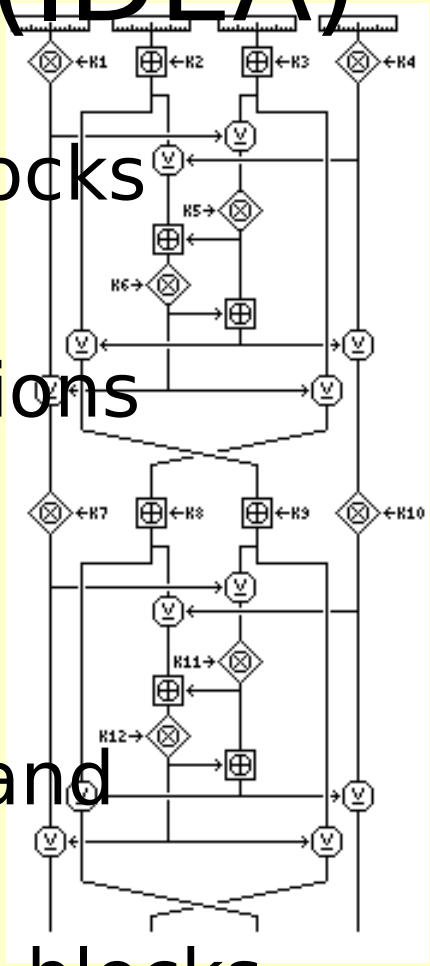
Shift Row

Mix Column

Add Round Key



International Data Encryption Algorithm (IDEA)



- Uses 64-bit input and output blocks
- Uses 128 bit key.
- Based on 3 mathematical functions
  - XOR
  - Addition modulo
  - Multiplication modulo
- Uses 8 rounds of transposition and substitution
- Implementation uses 16-bit sub blocks

RC6, Blowfish

RC6

- Data dependant rotations, variable block size, key size and number of rounds
- Submitted for AES contest – one of 5 finalists
- adds two features to RC5: inclusion of integer multiplication and the use of four 4-bit working registers instead of RC5’s two 2-bit registers.

Blowfish

- Key lengths ups to 448 bits
- 64 bit block size
- optimized for 32 bit microprocessors

Block Algorithms

Strength	Name	Key Size
Weak	RC2-40	40
	DES	56
	RC-5-64/16/7	56
Medium	RC5-64/16/10	80
	Skipjack	80
Strong	RC2-128	128
	RC5-64/12/16	128
	IDEA	128
	Blowfish	128
	3DES	168
Very Strong	RC5-64/12/32	256
	Twofish	256
	RC6	256
	Rindael	256

Encryption Protocols

- Encryption File System (EFS)
- Pretty Good Privacy (PGP)
- Defense Messaging System (DMS), Common Access Card (CAC), Fortezza Card
- Secure Socket Layer (SSL)
- Diffie-Hellman
- Windows NT Lan Manager (NTLM) Authentication
- UNIX passwords

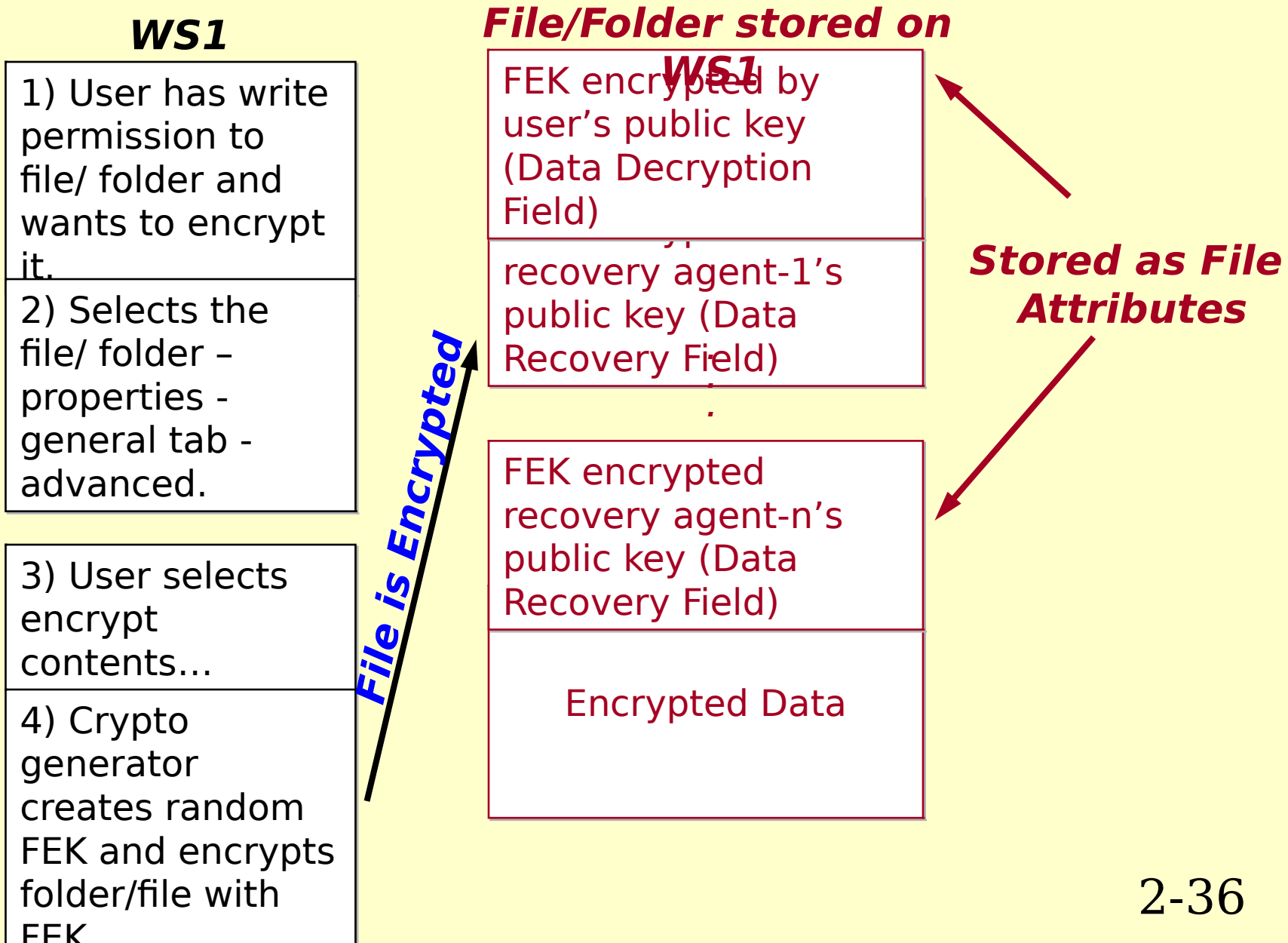
Encryption File System (1 of 2)

- Windows 2000 uses EFS (or Kerberos) to encrypt local files.
- EFS uses the DESX algorithm for file encryption.
  - Encryption key is 120 bits long
  - Called a FEK (file encryption key).
  - Encrypted with user's public key
- When a file (or folder) is encrypted:
  - FEK is generated.
  - FEK encrypts file or folder contents
  - FEK is stored encrypted as an attribute to the file
  - Uses a public key in both a data decryption field (DDF) and data recovery field (DRF).

EFS (2 of 2)

- Requires the use of certificates (public keys for both DRF and DDF).
- EFS must have:
  - a recovery agent available
  - file system must be NTFSv5
- Preferred method is to encrypt by folder:
  - less tedious
  - allows for automatic encryption
  - encrypt “My Documents” and “Temp” folders

Encrypting a File/Folder with EFS


WS1
1) User has write permission to file/ folder and wants to encrypt it.
2) Selects the file/ folder – properties - general tab - advanced.
3) User selects encrypt contents…
4) Crypto generator creates random FEK and encrypts folder/file with FEK
File is Encrypted
File/Folder stored on WS1
FEK encrypted by user's public key (Data Decryption Field)
recovery agent-1's public key (Data Recovery Field)
FEK encrypted recovery agent-n's public key (Data Recovery Field)
Encrypted Data
Stored as File Attributes

EFS – Things to Know

- A user with write permission can encrypt a file using their private key.
- If you attempt to send an encrypted file over the network it is decrypted by the crypto API and sent as plain text.
- Admin account is the default recovery agent.
- Do not encrypt files when you are logged in as the local admin (uses your key for DDF).
- Cannot encrypt files as root.
- Recommended: encrypt My Documents and Temp

Pretty Good Privacy (PGP)

- Uses Peer Trust instead of a CA (No CRL).
- Uses Rivest, Shamir, Adleman (RSA) as its public key algorithm (2048 bits).
- Secure; de facto standard in Europe.
- Uses IDEA (128 bit) for its conventional encryption algorithm.
- Digital signatures are available as are integrity checks using MD5 (for hashing).
- Major disadvantage no 3rd party checks.

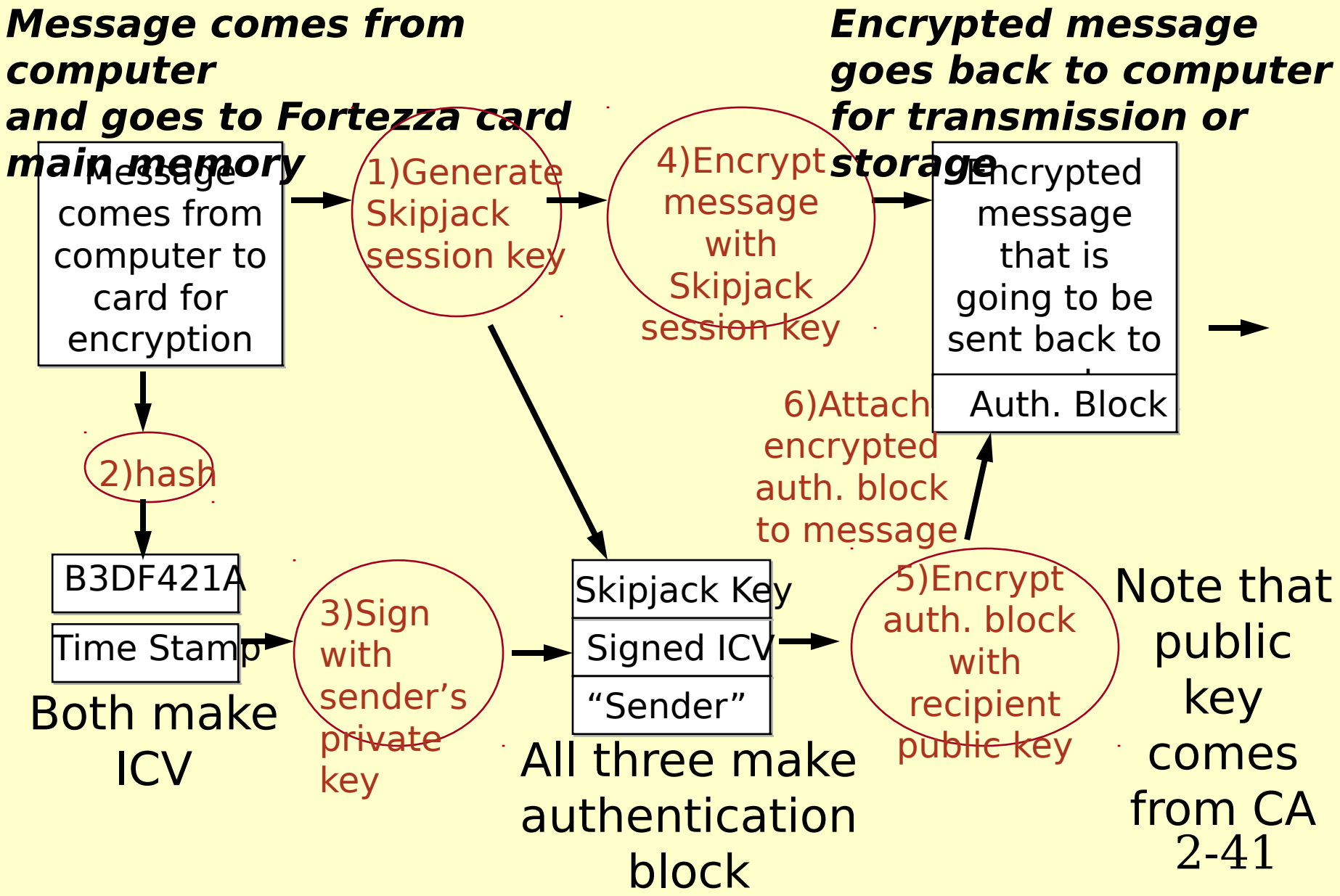
Defense Messaging System (DMS) and the CAC

	DMS	CAC
Hardware	Fortezza Card	Smart Card
Conventional Key Distribution	Key Encryption Algorithm	RSA Algorithm
Conventional Encryption	Skipjack (80 bits)	DES (56 bits), 3DES (112 bits)
Digital Signatures	Signed hash	Private Key used

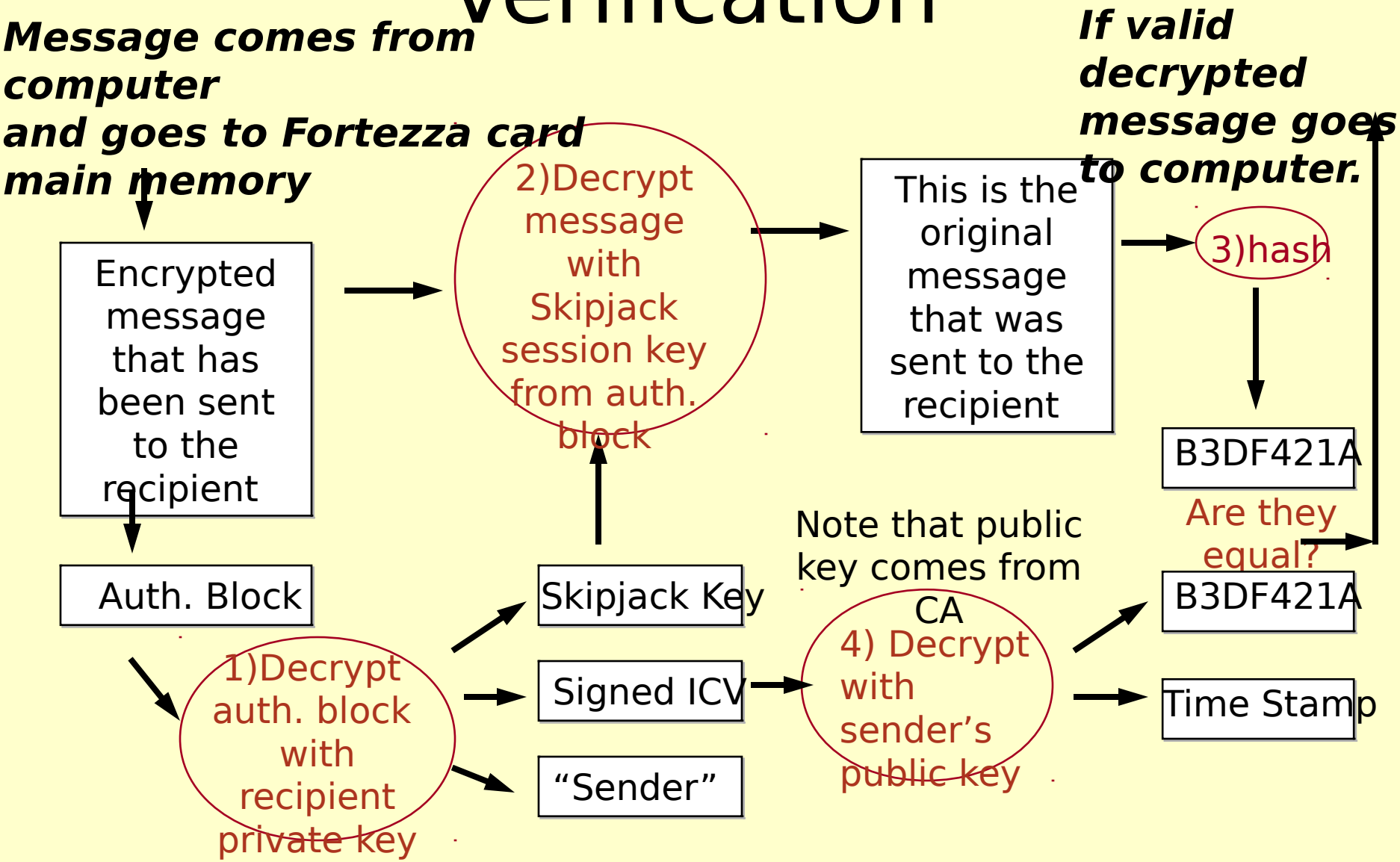
Fortezza Smart Cards

- Cryptographic module packaged on PCMCIA smart card.
- Fortezza cards have:
  - Clipper chip.
  - Clock for timestamps.
  - Permanent memory for certificates (public key storage).
  - Temporary registers for conventional key storage.
  - RAM for decryption and encryption of data.
- Fortezza cards are zeroed if:
  - Someone tampers with the card.
  - Attempt incorrectly to login 10 times in a row.

Fortezza Encryption and Signing


Message comes from computer and goes to Fortezza card main memory
Encrypted message goes back to computer for transmission or storage
Message comes from computer to card for encryption
1)Generate Skipjack session key
4)Encrypt message with Skipjack session key
Encrypted message that is going to be sent back to
Auth. Block
6)Attach encrypted auth. block to message
2)hash
B3DF421A
Time Stamp
Both make ICV
3)Sign with sender's private key
Skipjack Key
Signed ICV
"Sender"
All three make authentication block
5)Encrypt auth. block with recipient public key
Note that public key comes from CA

Fortezza Decryption & Verification


Message comes from computer and goes to Fortezza card main memory
Encrypted message that has been sent to the recipient
Auth. Block
1)Decrypt auth. block with recipient private key
Skipjack Key
Signed ICV
"Sender"
2)Decrypt message with Skipjack session key from auth. block
This is the original message that was sent to the recipient
Note that public key comes from CA
4) Decrypt with sender's public key
3)hash
B3DF421A
Are they equal?
B3DF421A
Time Stamp
If valid decrypted message goes to computer.

Secure Sockets Layer (SSL)

- Supports web browsers and servers
- Uses a one way public key algorithm (by default).
- Server sends its public key.
- Client generates a session key using browser.
- Client sends session key to Server encrypted with Server public key.
- Client and Server exchange info using session key generated by client.
- Can you be sure that the Server is who they claim?
- The standard case does not authenticate client.

Diffie-Hellman

- Asymmetric cipher
- Used for key exchange not encryption
- Invented in 1976 – first public key algorithm
- Patent expired in 1997
- Based on one-way function:
  - $Y^x \pmod P$
  - Sender/receiver agree on Y and P values
  - $Y < P$

Diffie-Hellman Key Exchange (1 of 2)

Alice	**Exchange Methodology**	**Bob**
Select Y and P with Bob	Y^x (mod P) (publicly known)	Select Y and P with Alice
Y = 11, P = 13	11^x (mod 13)	Y = 11, P = 13
Choose secret number (2)	Select secret number	Choose secret number (5)
11^2 (mod 13)	Calculate 1-way function using their number	11^5 (mod 13)

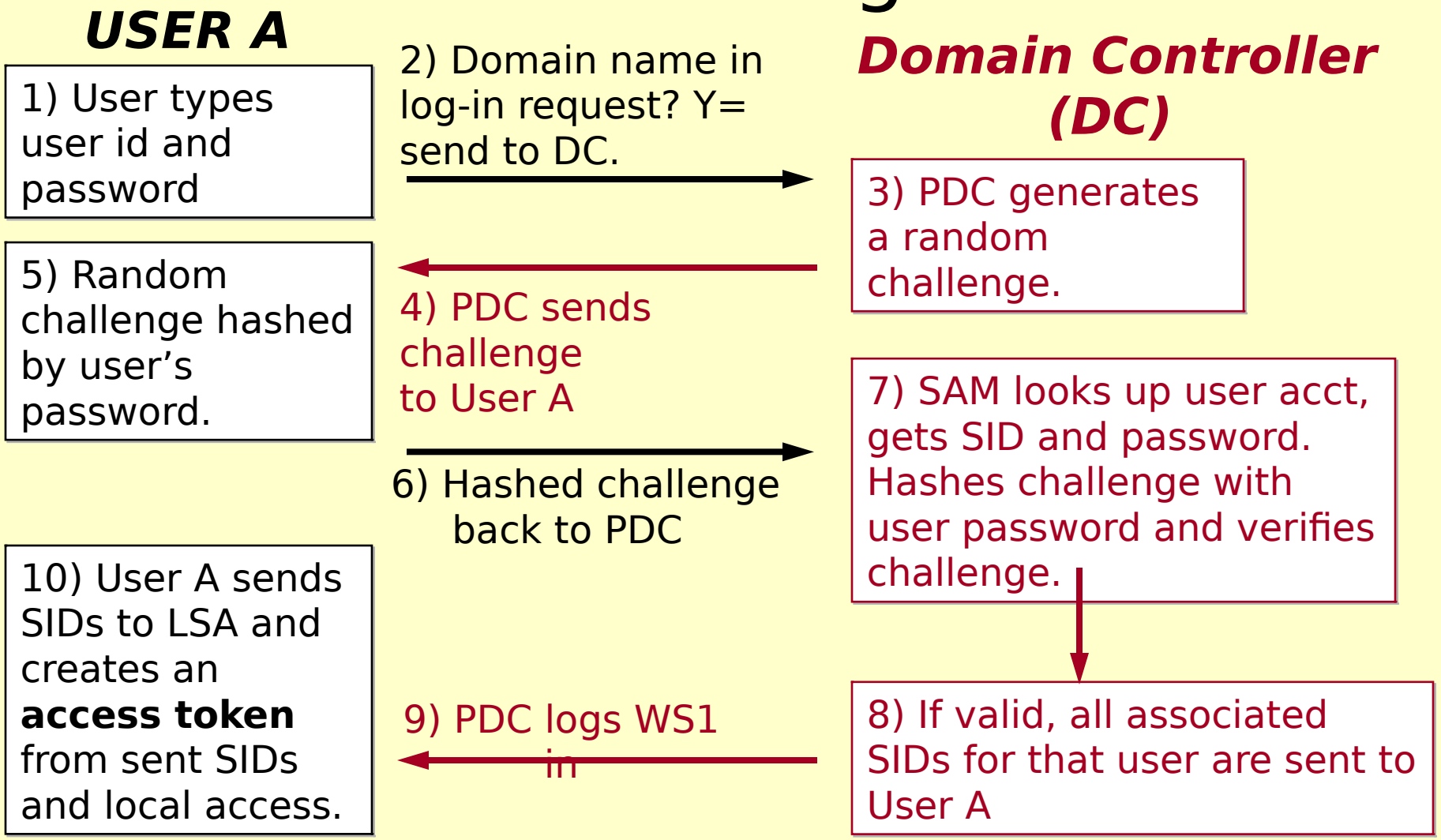
Diffie-Hellman Key Exchange (2 of 2)

Alice	**Exchange Methodology**	**Bob**
Sends “4” to Bob	Send the result of the 1-way function	Sends “7” to Alice
Calculate 7^2 (mod 13)	Use received number and raise it to your secret number	Calculate 4^5 (mod 13)
Secret key selected through	Both have same result w/o revealing	Secret key selected through

Challenge-Response NTLM

- User requests logon for a domain to PDC.
- PDC sends a random challenge to user’s WS.
- WS encrypts (hashes) challenge using the user password as the hash key value.
- Encrypted challenge sent back to PDC.
- PDC verifies challenge independently.
- If hashes match PDC sends all users SIDs back to WS.
- Local LSA takes SIDs checks local account data and then creates an access token.
- Winlogon launches user’s shell.

NTLM Challenge Response Domain Logon


USER A
Domain Controller (DC)
1) User types user id and password
2) Domain name in log-in request? Y= send to DC.
3) PDC generates a random challenge.
4) PDC sends challenge to User A
5) Random challenge hashed by user's password.
6) Hashed challenge back to PDC
7) SAM looks up user acct, gets SID and password. Hashes challenge with user password and verifies challenge.
8) If valid, all associated SIDs for that user are sent to User A
9) PDC logs WS1 in
10) User A sends SIDs to LSA and creates an **access token** from sent SIDs and local access.

NTLM – Things to Know

- Challenge and response can be captured by packet sniffer.
- Three hashes used, LanMan (LM), NTLM, NTLMv2.
- Default is both the NTLM and LM hash are sent.
- LM splits passwords into 7 character hashes, while NTLM or v2 use all 14 characters.
- LM is much easier to crack.
- Physical security of servers is critical.
- To set-up NTLMv2 all DCs must be NT 4.0 SP4+
- NTLMv2 can be 56 or 128 bits

Kerberos - Windows 2000

- An authentication service
  - Uses conventional encryption (only) default is DES
  - Based on users (not machines) identifying themselves on an untrusted network to access services in a distributed environment.
- Three parts to the exchange
  - Logon on to the domain (realm) through a KDC
  - Request a type of service
  - Use a service

Kerberos Step I
Logon to Network

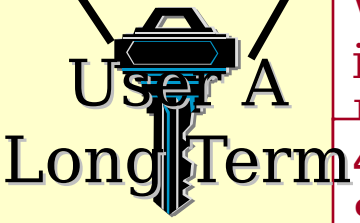
USER A

1) User enters logon and password. Kerberos client running on WS converts password to encryption key (called long-term key) and stores timestamp in credentials cache.

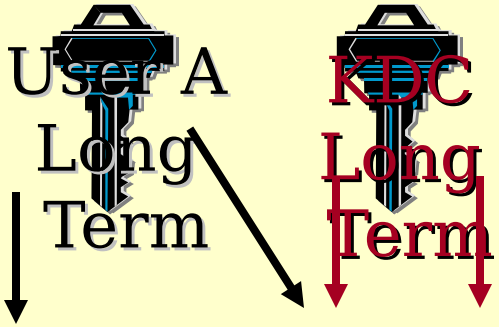
Key Distribution Center (KDC) or AS

User A + A's credentials

3) KDC – verifies User A by decrypting with User A's information in the KDC database



4) Creates a Log-on Session Key for User A. Encrypts Log-on Session Key with User A's long-term key.



5) Creates TGT which has another copy of User's A Session Key, User's A ID and authorization data.encrypted with KDC long-term key.

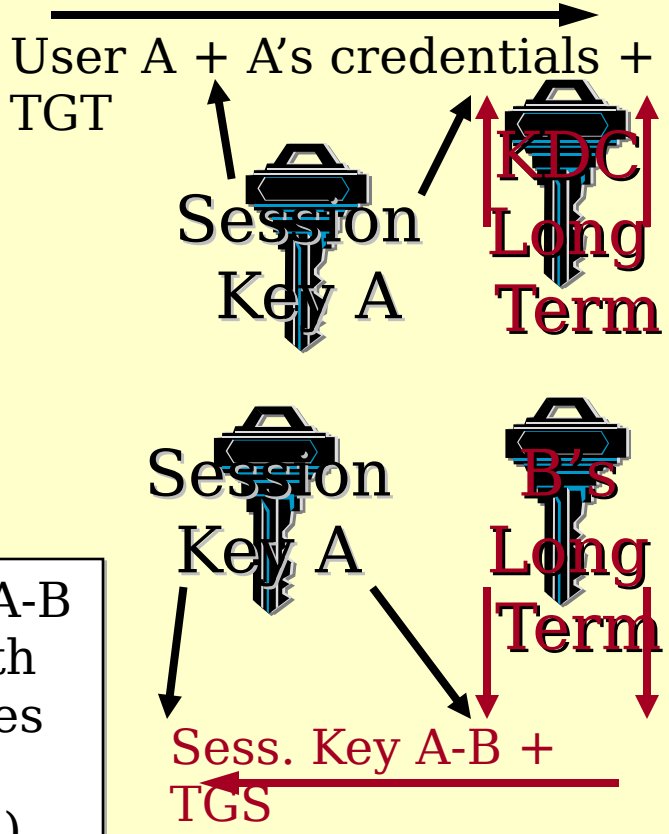
Session Key A + TGT

7) User A recovers session key (#4) by decrypting with long-term key, verifies timestamp and stores session key and TGT in credentials cache. Note that TGT cannot be decrypted.

Kerberos Step II
Get Access to Services

USER A (Wants to use SERVER B)

1) User A gets session key A in credentials cache and uses it to encrypt an authenticator and attaches the TGT and the service User A wants to use (Server B).



Ticket Granting Service

3) KDC – decrypts TGT with KDC's long-term key. Gets User A's session key A

4) User A.'s session key A used to decrypt A's credentials.

5) If valid. KDC checks authorization data in TGT. If valid it creates new session key called A-B. Encrypts A-B key with User A's Session key

6) KDC creates Session Ticket (called TGS) with key A-B, User A.'s auth. data & encrypts with Server B's long-term key..

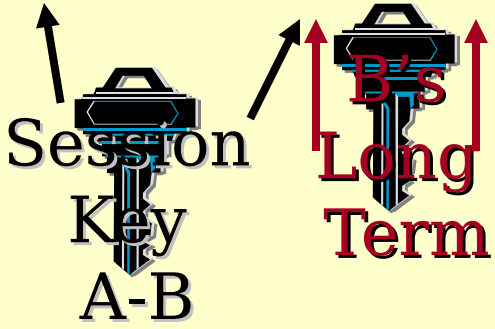
8) User A decrypts A-B session key (#5) with session key A. Stores session key A-B and Session Ticket (TGS) for Server B in credentials cache.

Kerberos Step III
Using a Specific Service

USER A

1) User A gets session key A-B in credentials cache and uses it to encrypt an authenticator for Server B adds TGS.

User A + A's credentials + TGS



SERVER B

3) Server B – decrypts Session Ticket with Server B's long-term key. Gets key A-B.

4) Key A-B used verify A's credentials.

timestamp

5) If valid.Server B takes the timestamp out of the authenticator and encrypts with Key A-B and sends back to User A.

7) User A decrypts timestamp with A-B session key and verifies it by checking its credentials cache. If valid connection proceeds. At this point all exchanged data with Server B can be encrypted with key A-B

8) Connection Proceeds

Kerberos – Things to Know

- Since timestamps are used in the authenticators - computer clocks must be within tolerance (default is five minutes).
- By default NTLM and Kerberos are loaded by the LSA on a Windows 2000 computer.
- The credentials cache is not paged and is erased upon logoff or system shut-down
- Smart Cards can be used to replace the password logon.
- Kerberos is authentication NOT authorization.

Encryption

- Attacks
- Network Traffic
  - Link Encryption
  - End-to-end Encryption
  - Key Management
- Certificate Authorities
- Encryption Algorithms
- Encryption v. Password Strength
- Computer Network Defense

UNIX Passwords

- Normal encryption of password file uses DES (56 bits) or a similar algorithm.
- Several UNIX systems send login and password in plaintext b/w client/server.
- Use OPIE – (one time password for everything) or a similar program to increase security.

Encryption vs Password Strength

- /etc/shadow
- common encryption algorithms (weak)
- packet sniffers
- Difference between a complex and a dictionary password
  - is 2,000,000 vs 218,000,000,000,000 (109,000,000 times).
  - dictionary takes 2 seconds, complex takes 6.91 years
  - LANMAN hash makes it 78,000,000,000 or 1.85 days

Encryption

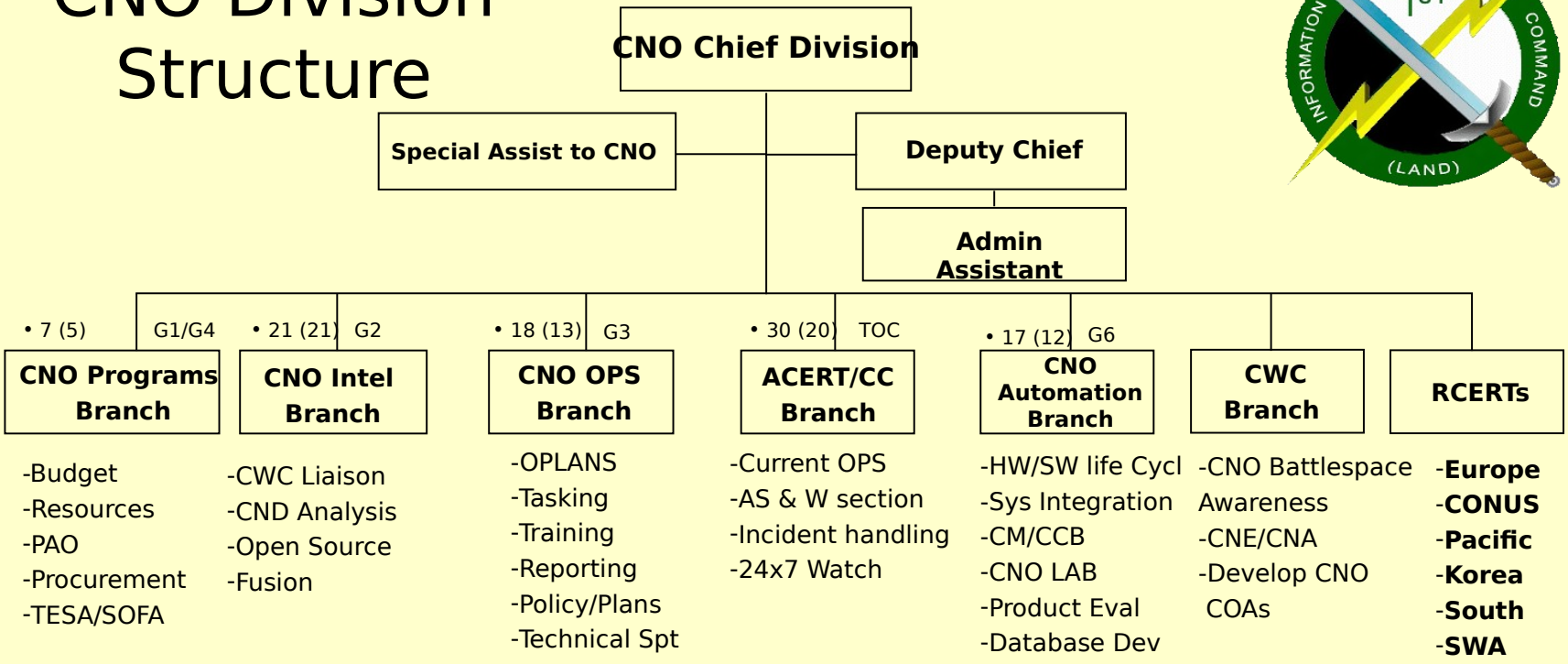
- Attacks
- Network Traffic
  - Link Encryption
  - End-to-end Encryption
  - Key Management
- Certificate Authorities
- Encryption Algorithms
- Encryption v. Password Strength
- Computer Network Defense

Army Computer Emergency Response Team (ACERT)

- Provides Computer Defense Assistance Program (CDAP)
- Provides Information Assurance Vulnerability Assessments (IAVA)
- Serves as clearinghouse for network security problems

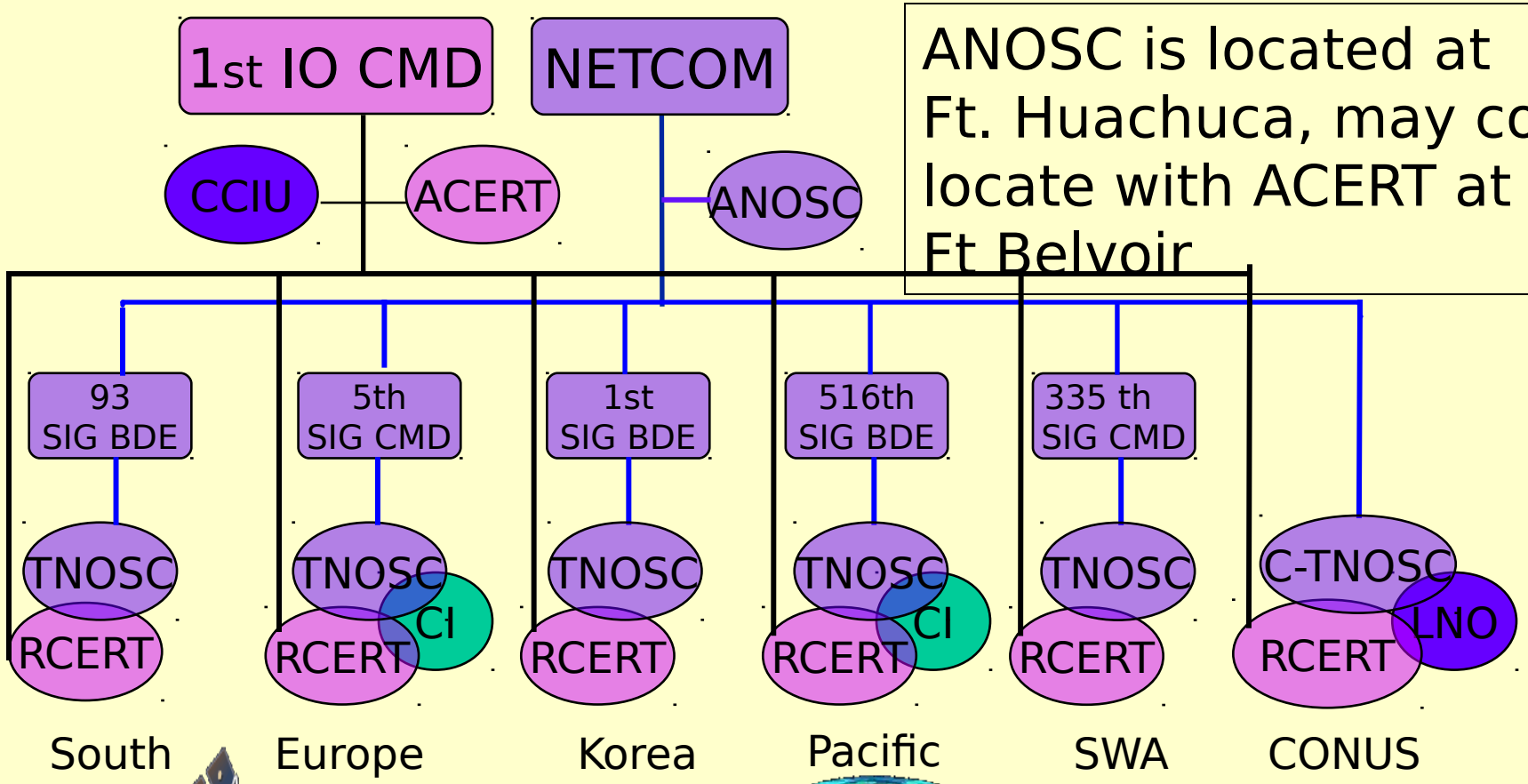
ACERT

CNO Division Structure



Mission: Conducts Computer Network Defensive operations in support of the U.S. Army to ensure the confidentiality, integrity, and availability of the information and information systems used by commanders world wide

World Wide Concept



Computer Defense Assistance Program

Mission:

- Assist Army Commands/Activities with computer and network security

Objectives:

- Focus on network configuration (Security)
- Identify potential points of unauthorized access
- Validate vulnerabilities and assess depth and degree of potential compromise
- Recommend methods, techniques, and configuration modifications needed to secure the networks

Authorization:

- C2 Protect Implementation Plan
- LIWA/ACERT Charter - DCSOPS, DISC4, DCSINT IO Triad
- AR 25-2, AR 380-53, NTISSD 600, and DoD Directive 4640.1

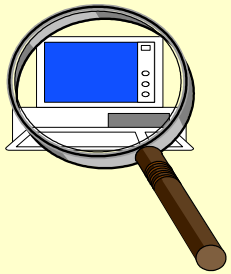
CDAP Process

CDAP STEPS 1 & 2
(Coordination / Authorization)

Commander Request

CDAP STEPS 3 & 4
(Mapping / Scanning)

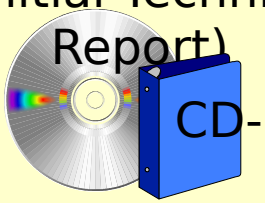
CDAP STEP 5 & 6
(Penetration and Verification Tests (optional)



CDAP STEP 7
(Assistance)

- Assist with countermeasu
and configuration fixes

(Initial Technical Report)



CDAP STEP 8
(Final Report)

Final Report on CD-ROM

OUT BRIEF

DITY VAP Mission

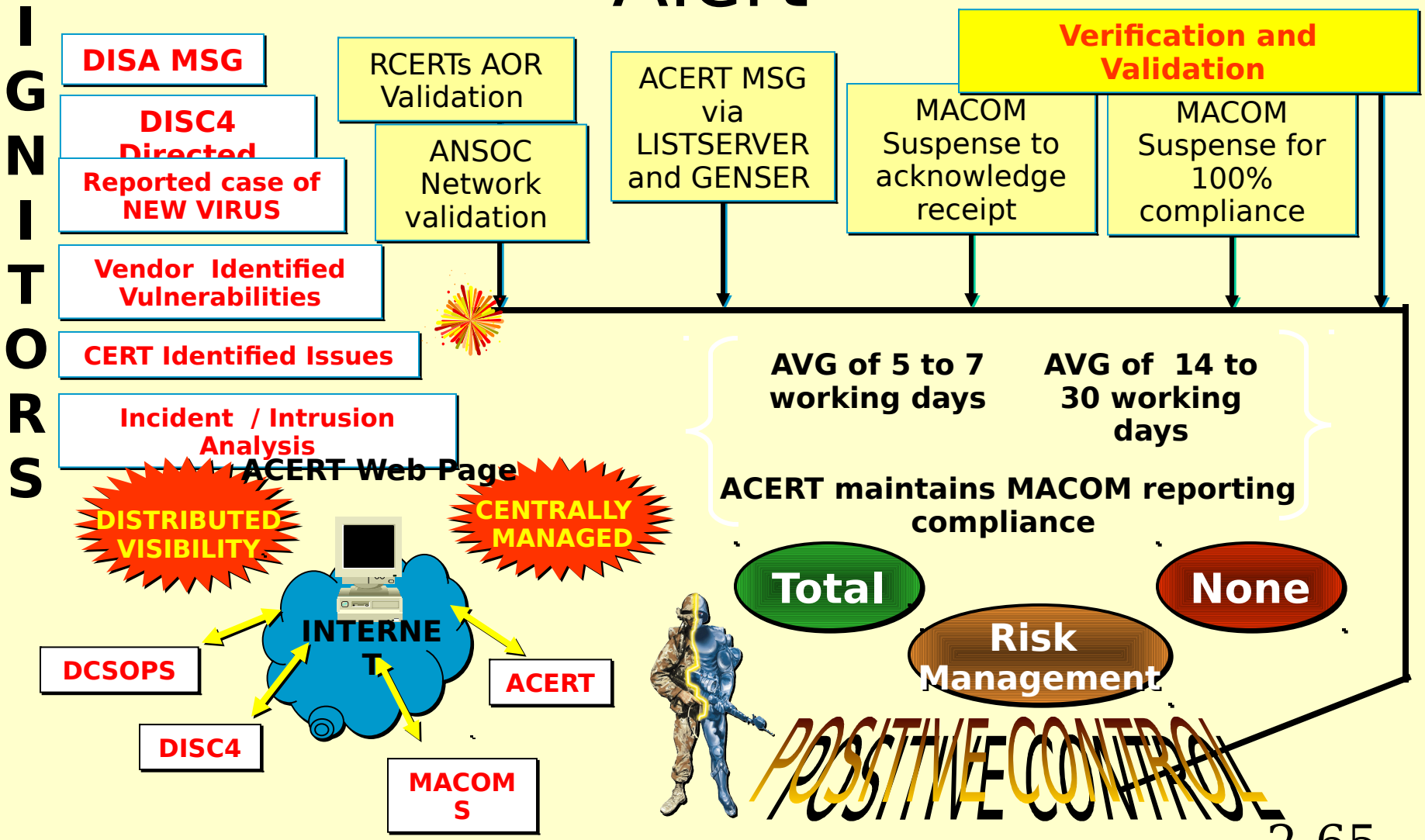
Mission:

- Assist Army Commands/Activities with IS security

Objectives:

- Provide organic unit capability for self-assessment
- Create a library of assessment tools for checkout
- Provide training and technical support
- Push security awareness down to appropriate level
- Control/limit scans across the infrastructure
- Recommend methods, techniques, and configuration modifications needed to secure the networks

Info Assurance Vulnerability Alert



Summary

- You need to know what forms of encryption your operating system uses and HOW IT WORKS and HOW STRONG IT IS – it defines your problems.
- Both Windows 2000 and Solaris 8.0 have multiple forms of encryption and can integrate COTS encryption mechanisms.
- Logon processes affect the placement of domain controllers, certificate authorities and the distribution of certificates.
- Physical security of servers is very critical.